This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2014.2358031, IEEE Transactions on Circuits and Systems for Video Technology

Efficient Feature Selection and Classification for Vehicle Detection

Xuezhi Wen, Ling Shao, *Senior Member, IEEE,* Wei Fang and Yu Xue

***Abstract*—The focus of this work is on the problem of Haar-like feature selection and classification for vehicle detection. Haar-like features are particularly attractive for vehicle detection because they form a compact representation, encode edge and structural information, capture information from multiple scales, and especially can be computed efficiently. Due to the large-scale nature of the Haar-like feature pool, we present a rapid and effective feature selection method via AdaBoost by combining a sample's feature value with its class label. Our approach is analyzed theoretically and empirically to show its efficiency. Then an improved normalization algorithm for the selected feature values is designed to reduce the intra-class difference while increasing the inter-class variability. Experimental results demonstrate that the proposed approaches not only speed up the feature selection process with AdaBoost but also yield better detection performance than the state-of-the-art methods.**

***Index Terms*—Haar-like features, SVM, AdaBoost, weak classifier, vehicle detection.**

I. INTRODUCTION

Vision-based vehicle detection for driver assistance has received considerable attention over the last 15 years. There are at least three reasons for the booming research in this field: 1) the startling losses both in human lives and finance caused by vehicle accidents, 2) the availability of feasible technologies accumulated within the last 40 years of computer vision research, and 3) the exponential growth in processor speeds that have paved the way for running computation-intensive video-processing algorithms even on a low-end PC in real-time. On-board vehicle detection systems have high computational requirements as they need to process the acquired images in real-time or close to real-time for instant driver reaction. Searching the whole image to locate potential vehicle locations is prohibitive for real-time applications. The majority of methods reported in the literature follow two basic steps: 1) hypothesis generation (HG) where the locations of possible vehicles in an image are hypothesized and 2) hypothesis verification (HV) where tests are performed to verify the presence of vehicles in an image (see Fig. 1).

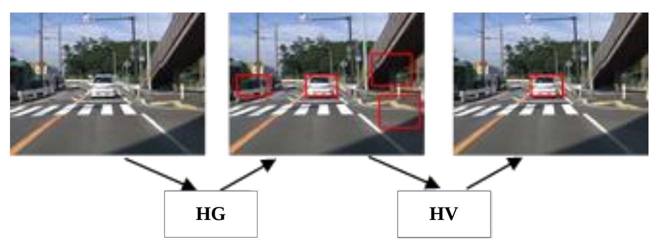


Fig. 1. Vehicle detection process.

The input to the HV step is the set of hypothesized locations from the HG step. During HV, tests are performed to verify the correctness of a hypothesis. Approaches to HV can be classified mainly into two categories: 1) template-based and 2) appearance-based. Template-based methods use predefined patterns from the vehicle class and perform correlation. Appearance-based methods, which are also called machine learning methods, on the other hand, learn the characteristics of the vehicle class from a set of training images which should capture the variability in vehicle appearance. Usually, the variability of the non-vehicle class is also modeled to improve the performance.

The HV step acts as an important role for vehicle detection. Template-based methods need to use thousands of predefined patterns of the vehicle class and perform correlation between the test image and the template, which makes them time-consuming. In addition, template-based methods are sensitive to the varying background (e.g., buildings, bridges

This work was supported by the Jiangsu Planned Projects for Postdoctoral Research Funds under Grant 1102108C, the National Natural Science Foundation of China under Grant 61403206, the Natural Science Foundation of Jiangsu Province under Grant BK20141005, the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 14KJB520025, the Research project of Nanjing University of Information Science and Technology under Grant 20110434, the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), and the Open Research Project of State Key Laboratory of Novel Software Technology under Grant KFKT2014B21. (*Corresponding author: Ling Shao.*)

X. Wen is with Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China. (e-mail: ww_pub@163.com).

L. Shao is with the Department of Computer Science and Digital Technologies, Northumbria University, Newcastle upon Tyne, NE1 8ST, U.K. (e-mail: ling.shao@ieee.org).

W. Fang is with Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China, and with School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with State Key Laboratory for Novel Software Technology, Nanjing University, P.R. China (e-mail: hsfangwei@sina.com).

Y. Xue is with Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China. (e-mail: xueyu@nuist.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2014.2358031, IEEE Transactions on Circuits and Systems for Video Technology

and guardrails). Therefore, the appearance-based validation approaches have become more attractive. There are at least two fundamental challenges faced by the appearance-based validation methods: the processing time and accuracy.

In this paper, we focus on the investigation of the appearance-based validation approaches to HV. Seeking the solutions to boost the vehicle detection accuracy and reduce the false alarm rate while considering the real time, we propose a machine learning algorithm based on Haar-like features and SVM. Specifically, we first design a Haar-like features extraction method to represent a vehicle's edges and structures, and then propose a rapid feature selection algorithm using AdaBoost due to the large pool of Haar-like features. Finally, we present an improved normalization method for feature values. Experimental results demonstrate that the proposed approaches not only speed up the feature selection process with AdaBoost but also outperform the state-of-the-art methods in terms of classification ability.

The rest of the paper is organized as follows. In Section II, we review the related work for vehicle detection by using appearance-based approaches. In Section III, we present an algorithm for computing Haar-like features. A fast feature selection method based on AdaBoost is reported in Section IV. Section V gives an introduction of SVMs and introduces an improved normalization method for the original feature values while training SVM. The experimental results and analysis are described in Section VI. Section VII concludes this paper.

II. Related Work

Machine learning methods are becoming increasingly popular for their high performance, good robustness and easy operation, which have been applied to many fields (such as image retrieval, image annotation, visual recognition and vehicle detection) [1]-[4]. HV using machine learning methods is treated as a two-class pattern classification problem: vehicle versus non-vehicle. In general, machine learning methods consist of two processes: (1) feature representation and (2) classification.

A. Feature representation

Given the huge intra-class variabilities of the vehicle class, one feasible approach is to learn the decision boundary based on training a classifier using the feature sets extracted from a training set. Various feature extraction methods have been investigated in the context of vehicle detection. Based on the method used, the features extracted can be classified as either global or local.

Global features are obtained by considering all the pixels in an image. Usually dimensionality reduction techniques [5] [6] are required for the high-dimensional features. Wu and Zhang [7] used standard Principal Component Analysis (PCA) for feature extraction, together with a nearest-neighbor classifier, reporting an 89 percent accuracy on a vehicle dataset. However, their evaluation database was quite small (93 vehicle images and 134 non-vehicle images), which makes it difficult to draw any useful conclusions. Although detection schemes based on global features such as those described in [7]-[13] perform reasonably well, an inherent problem with global feature extraction approaches is that they are sensitive to local or global image variations (e.g., viewpoint changes, illumination changes, and partial occlusion).

Local features, on the other hand, are less sensitive to the effects faced by global features. Moreover, geometric information and constraints in the configuration of different local features can be utilized either explicitly or implicitly. An overcomplete dictionary of Haar wavelet features was utilized in [14] for vehicle detection. They argued that this representation provided a richer model and spatial resolution and that it was more suitable for capturing complex patterns. Sun et al. went one step further by arguing that the actual values of the wavelet coefficients are not very important for vehicle detection. They proposed using quantized coefficients to improve detection performance [15]. Using Gabor filters for vehicle feature extraction was investigated in [16]. Gabor filters [17] provide a mechanism for obtaining orientation and scale related features. The hypothesized vehicle subimages were divided into nine overlapping subwindows, and then Gabor filters were applied on each subwindow separately. Furthermore, Sun et al. [18] combined Haar wavelet with Gabor features to describe the properties of a vehicle. Scale invariant feature transform (SIFT) features [19] were used in [20] to detect the rear faces of vehicles. In [21], the histogram of oriented gradients (HOG) features were extracted in a given image patch for vehicle detection. In [22], a combination of speeded up robust features (SURF) [23] and edges was used to detect vehicles in the blind spot.

The main drawback of the above local features is that they are quite slow to compute. In recent years, there has been a transition from complex image features such as Gabor filters and HOG to simpler and efficient feature sets for vehicle detection. Haar-like features are sensitive to vertical, horizontal, and symmetric structures, and they can be computed efficiently, making them well suited for real-time detection of vehicles [24], also demonstrated by their good performance in the object detection literature [25]-[27]. Accordingly, we choose Haar-like features as the feature representation for our vehicle detection system.

B. Classification

Classification can be broadly split into two categories: discriminative and generative methods. Discriminative classifiers, which learn a decision boundary between two classes, have been more widely used in vehicle detection. Generative classifiers, which learn the underlying distribution of a given class, have been less common in the vehicle detection literature. While in [28] and [29] artificial neural network classifiers were used for vehicle detection, they have recently fallen somewhat out of favor. Neural networks have many parameters to tune, and the training tends to converge to a local optimum. The research community has moved toward classifiers whose training converges to a global optimum over the training set, such as SVMs and AdaBoost. SVMs have been widely used for vehicle detection. In [30] and [31], SVM was

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2014.2358031, IEEE Transactions on Circuits and Systems for Video Technology

used to classify feature vectors consisting of Haar wavelet coefficients. The combination of HOG features and the SVM classifier has been also used in [32], [33] and [28]. The HOG-SVM formulation was extended to detect and calculate vehicle orientation using multiplicative kernels in [34]. Edge features were classified for vehicle detection using SVM in [35] and [36]. In [37], vehicles were detected using Haar and Gabor features, using SVM for classification. AdaBoost has been also widely used for classification, largely owing to its integration with cascade classification in [25]. In [38], AdaBoost was used for detecting vehicles based on symmetry feature scores. In [39], edge features were classified using AdaBoost. The combination of Haar-like feature extraction and AdaBoost classification has been used to detect rear faces of vehicles in [40]–[44].

In addition, Szegedy et al. [45] defined a multi-scale inference procedure which is able to produce high-resolution object detectors based on deep neural networks (DNNs).

Compared with the popular AdaBoost classifiers, SVM is slower in the test stage. However, the training of SVM is much faster than that of AdaBoost classifiers. Similarly, although DNNs can yield strong results for object detection, these results come at heavy computational costs during training. Therefore, we choose SVM as the classifier in this paper.

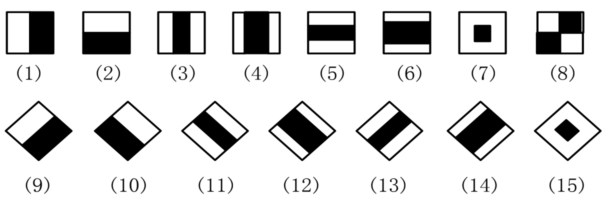


Fig. 2. Haar-like feature prototypes used in our method: upright ones (the first row), rotated ones (the second row).


Fig. 3. Haar-like feature examples for describing a vehicle's appearance.

III. Haar-like Feature extraction

Rather than using pixels, Viola et al. [25]-[27] used simple Haar feature prototypes to extract features to encode an image patch for human face detection (the first row of Fig. 2). To further lower the false alarm rate at a given hit rate, Lienhart et al. [46] [47] introduced new Haar feature prototypes by rotating these simple ones by 45 degrees (the second row of Fig. 2), and the results proved to be effective. Hence we take all of these simple and rotated prototypes. And we speed up the feature extraction procedure using an intermediate representation for the image patch -- integral image (see [25] for details). Fig. 3 gives a few examples of Haar-like features for the description of a vehicle's appearance [24].

For a given image, the region of interest (ROI), i.e., the vehicle region, is segmented using shadow, symmetry and aspect ratio information according to [30] [31]. Considering the property of the structure of a vehicle, we add the diagonal features which are described in [46] [47], and the whole Haar-like feature pool we deploy is summarized in Table I. The procedure for computing the Haar-like feature pool is shown in Algorithm 1.

TABLE I
THE NUMBER OF FEATURES FOR AN IMAGE PATCH WITH SIZE OF 32×32

Feature type	w/h	Feature number
(1); (2)	2/1; 1/2	13,904
(9); (10)	2/1; 1/2	7,260
(3); (5)	3/1; 1/3	9,570
(4); (6)	4/1; 1/4	7,176
(11); (13)	3/1; 1/3	5,184
(12); (14)	4/1; 1/4	3,944
(7); (15)	3/3; 3/3	5,025
(8)	2/2	5,456
Total		57,519

Algorithm 1 Computing the Haar-like feature pool

Input
A ROI image patch in RGB color space
Begin
1) Normalize ROI to 32×32 in grayscale
2) Compute the upright and rotated integral images
3) Compute all Haar-like feature values with the integral images according to Table I
End begin
Output
Haar-like feature pool

IV. Feature selection

The scale of the obtained Haar-like feature pool is far more than the pixels of a 32×32 gray scale image. Even though each feature can be quickly computed, the whole process is still quite time-consuming. In fact, only a few features among them play an important role for classification, which can be regarded as key features. The AdaBoost algorithm is an effective way to select these key features. The traditional feature selection and the proposed one via AdaBoost are detailed respectively as follows.

A. *Traditional feature selection*

The traditional feature selection process with the AdaBoost algorithm is illustrated in Algorithm 2 according to [48].

Algorithm 2 The AdaBoost algorithm for feature selection

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2014.2358031, IEEE Transactions on Circuits and Systems for Video Technology

Input

1) A training set:

$$\{x_i, y_i\}_{i=1}^{n}, x_i \in X, y_i \in \{-1,+1\}, i = 1,2,\cdots,n$$

where n is the size of the training set

2) x_i denotes the feature vector of the *i*th sample

3) y_i denotes the class label of the *i*th sample

4) X denotes the feature space

Begin

1) Initialize weights: $w_1(i) = 1/n \quad i = 1,2,\cdots,n$

2) $H = null$ // // Key feature set

3) For t = 1 to T

(1) Normalize the weights:

$$w_t(i) = w_t(i) / \sum_{i=1}^{n} w_t(i) \quad i = 1,2,\cdots,n$$

(2) For each feature j, train a weak classifier f_j.

(3) The error ε_j of a classifier f_j is evaluated as follows:

$$\varepsilon_j = \sum_{i=1}^{n} w_{t,i} \kappa(x_i)$$

where $\kappa(x_i) = \begin{cases} 0 & f_j(x_i) = y_i \\ 1 & else \end{cases}$

(4) Choose the classifier f_t with the lowest error ε_t and $H = H \cup \{t\}$

(5) Compute $\alpha_t = \frac{1}{2}\ln((1-\varepsilon_t)/\varepsilon_t)$.

(6) Update the weights:

$$w_{t+1}(i) = w_t(i) * \exp(-\alpha_t f_t(x_i) y_i))$$

End for

4) $F(x) = sign(\sum_{t=1}^{T} \alpha_t f_t(x))$

End begin

Output

1)Key feature set H

2)AdaBoost classifier $F(x)$

From Algorithm 2, we can find that the time of feature selection is mostly consumed on finding the weak classifiers. In general, at each iteration, generating weak classifiers consists of three stages considering each feature: (1) generate the latent classification locations, (2) compute the classification error on each latent classification location, and (3) select the best classifier (weak classifier) which has the lowest error.

B. Proposed feature selection

The difference between the proposed feature selection method and the traditional one lies in stage (1): the traditional method only uses the feature values to generate the latent classification locations, and the proposed approach generates the latent classification locations by combining the feature values with their class labels. Without loss of generality, Fig. 4 presents an example of the difference between the two methods for the given ten feature values.

For the traditional method, it uses the exhaustive method to generate the latent classification locations. Specifically, it takes the middle location of every two adjacent feature values as the latent classification location, whereas the proposed method takes the class labels into account, i.e., only the middle location of the two adjacent feature values with different labels is considered as the latent classification location.

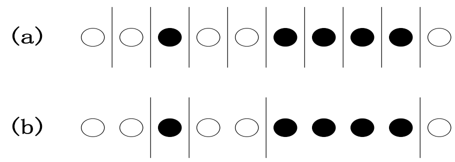


Fig. 4. The difference between the traditional feature selection method and the proposed approach. The hollow and solid circles denote two different classes respectively. (a) The traditional feature selection method. (b) The proposed feature selection approach.

C. Theoretical analysis for the proposed approach

In the last subsection, we have presented the proposed feature selection method which combines the feature values with their class labels. In this subsection, we theorectically analyze our approach in terms of the property of the class labels. For convenience, we assume l is the latent classification location, and the classification results of the left of l are $\lambda(\lambda \in \{-1,+1\})$; on the contrary, the results of the right of l are $-\lambda$, and the classification error ε can be computed as Eq. (1).

$$\varepsilon = \frac{1}{4}\sum_{j=1}^{n} w_j \left(f(x_j) - y_j\right)^2) \tag{1}$$

where w_j is the *j*th sample's weight, $f(x_j) \in \{-1,+1\}$ is the classification result on the *j*th sample, and $y_j \in \{-1,+1\}$ is the real class label of the *j*th sample. So

$$\begin{aligned} \varepsilon &= \frac{1}{4}(\sum_{j=1}^{n} w_j \left(f(x_j) - y_j\right)^2) \\ &= \frac{1}{4}(\sum_{j=1}^{l-1} w_j \left(\lambda - y_j\right)^2 + \sum_{j=l+1}^{n} w_j \left(-\lambda - y_j\right)^2) \\ &= \frac{1}{4}\sum_{j=1}^{n} w_j \left(\lambda^2 + y_j^2\right) + \frac{1}{2}\lambda\left(\sum_{j=l+1}^{n} w_j y_j - \sum_{j=1}^{l-1} w_j y_j\right) \end{aligned} \tag{2}$$

As w_j and y_j are known, $\sum_{j=1}^{n} w_j y_j$ is also known. From $\sum_{j=1}^{n} w_j y_j = \sum_{j=1}^{l-1} w_j y_j + \sum_{j=l+1}^{n} w_j y_j$ and $\lambda^2 = y_j^2 = 1, \sum_{j=1}^{n} w_j = 1$,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2014.2358031, IEEE Transactions on Circuits and Systems for Video Technology

we can compute ε as follows:

$$\begin{aligned}\varepsilon &= \frac{1}{4}\sum_{j=1}^{n} w_j\left(\lambda^2 + y_j^2\right) + \frac{1}{2}\lambda\left(\sum_{j=1}^{n} w_j y_j - 2\sum_{j=1}^{l-1} w_j y_j\right) \\ &= \frac{1}{2} + \frac{1}{2}\lambda\left(\sum_{j=1}^{n} w_j y_j - 2\sum_{j=1}^{l-1} w_j y_j\right) \end{aligned} \quad (3)$$

Let's discuss the different cases of λ.

(1) When $\lambda = 1$, (3) turns into (4):

$$\varepsilon = \frac{1}{2} + \frac{1}{2}\left(\sum_{j=1}^{n} w_j y_j - 2\sum_{j=1}^{l-1} w_j y_j\right) \quad (4)$$

So finding $\min(\varepsilon)$ is to compute $\max(\sum_{j=1}^{l-1} w_j y_j)$. As $w_j > 0$, only when $y_{l-1} = 1$ and $y_{l+1} = -1$, does $\sum_{j=1}^{l-1} w_j y_j$ reach the maximum.

(2) When $\lambda = -1$, (3) turns into (5):

$$\varepsilon = \frac{1}{2} - \frac{1}{2}\left(\sum_{j=1}^{n} w_j y_j - 2\sum_{j=1}^{l-1} w_j y_j\right) \quad (5)$$

So finding $\min(\varepsilon)$ is to compute $\min(\sum_{j=1}^{l-1} w_j y_j)$. Only when $y_{l-1} = -1$ and $y_{l+1} = 1$, does $\sum_{j=1}^{k_{s_t}} w_j y_j$ reach the minimum.

The above analysis demonstrates that our proposed feature selection method by combining feature values with their class labels is reasonable and effective.

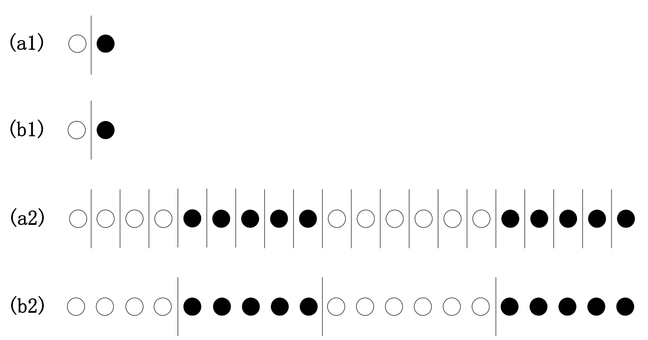


Fig. 5. An example for the analysis of the adaptability of the proposed approach with the number of the training samples as 2 and 20 respectively. The hollow and solid circles denote two different classes respectively. (a1) and (a2) denote the traditional feature selection method. (b1) and (b2) denote our proposed feature selection approach.

D. Scalability

According to the above theoretical analysis, the scalability of the proposed approach is further illustrated in Fig. 5 --- the traditional feature selection method need to compute the classification errors of 19 latent classification locations, while the proposed method computes the classification errors of only 3 latent classification locations, which saves much training time. The larger scale the training dataset is, the more consumed training time the proposed appoach saves. Therefore, the proposed method would be more advantageous for a larger number of training samples.

V. SVM CLASSIFIER

SVMs are primarily two-class classifiers that have been shown to be an attractive and more systematic approach to learning linear or non-linear decision boundaries [49] [50]. If the training examples from two classes cause the two classes' margin to be maximal, then the classification hyperplane satisfies the following equation:

$$f(x) = \sum_{i=1}^{m} y_i a_i k(x, x_i) + b \quad (6)$$

where $x, x_i \in \Re^n$ are n-dimensional input feature vectors, m is the number of examples, $y_i \in \{-1,+1\}$ is the label of the ith example, and $k(x, x_i)$ is a kernel function. We use the radial basis function (RBF) as the kernel function which is defined as:

$$k(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\delta^2}\right) \quad (7)$$

A. Data normalization

Data normalization is an essential step for most object detection algorithms that learn the statistical characteristics of attributes extracted from the object images, which can effectively reduce the within-class variation and increase the between-class variability. Data normalization is to scale the values of each continuous attributes into a well-proportioned range such that the effect of one attribute cannot dominate the others. A statistical normalization method was used in [51] and [52] to convert the data into a standard normal distribution while a min-max normalization method was adopted in [53] to directly convert the data into a range of 0 and 1.

The statistical normalization is defined as:

$$x_i' == \frac{x_i - \mu}{\sigma} \quad (8)$$

where μ is the mean of n values for a given attribute, and σ is its standard deviation. However, by using the statistical normalization, the data set should follow a Normal distribution.

The min-max normalization is defined as:

$$x_i' == \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \quad (9)$$

Normally x_i' is set to zero if the maximum is equal to the minimum. However, the min-max normalization method is sensitive to the lighting condition if it is directly used to the image data.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2014.2358031, IEEE Transactions on Circuits and Systems for Video Technology

In order to overcome the problem faced by the min-max normalization, we present an improved normalization algorithm based on the min-max normalization method. The main idea follows this observation: the actual feature values are not very important for vehicle detection. In fact, the magnitudes indicate local oriented intensity differences, and this information could be very different even for the same vehicle under different lighting conditions. The proposed method firstly computes the magnitudes of the obtained feature values, which is the main difference from the traditional methods, and then normalizes the magnitudes to [0, 1] by using the min-max method. The detailed process is presented in Algorithm 3.

Algorithm 3 The improved normalization algorithm

Input

A training set $\{x_i, y_i\}_{i=1}^{n}(y_i \in \{-1,+1\})$,

where $x_i = (v_{i1}, \cdots, v_{il})^T, (l << m)$

Begin

For $j = 1$ to l

For $k = 1$ to n

Compute the absolute value: $\left|v_{jk}\right|$

End for

max_value = $\max\{\left|v_{jk}\right|_{k=1}^{n}\}$

min_value = $\min\{\left|v_{jk}\right|_{k=1}^{n}\}$

For $k = 1$ to n

$v_{jk}' = \left(\left|v_{jk}\right| - \min\_value\right) / \left(\max\_value - \min\_value\right)$

End for

End for

End begin

Output

The normalized feature vector set:

$x_i' = (v_{i1}', \cdots, v_{il}')^T, i = 1, \cdots, n$

B. Training process

After performing the improved normalization operation, all feature values are normalized to $[0,1]$. Then the normalized feature vector set is used to train the RBF-SVM classifier with cross-validation to select the optimal parameters: σ and C .

C. Testing process

For a given test ROI image patch, we first normalize it to a 32×32 grayscale patch, and then compute the feature values according to the selected Haar-like features and normalize the feature values to $[0,1]$ according to the improved normalization algorithm shown in Algorithm 3. Finally, we construct the normalized feature values to a vector and input it to the trained RBF-SVM classifier, and then obtain the classification result.

(a) Vehicle samples.

(b) Non-vehicle samples.

Fig. 6. Examples of training images.

(a) Vehicle samples.

(b) Non-vehicle samples.

Fig. 7. Examples of test images of Test data II.

VI. Experimental Results and Analysis

To evaluate the proposed approaches, we apply them to a monocular-vision based detection system for static rear-vehicle images. This system includes two modules. The first module aims to segment ROIs accurately according to [30] [31]. The second module, which is the focus of this paper, performs classification on the ROIs. Vehicle existence validation is a two-class pattern classification problem: vehicle vs. non-vehicle.

Different videos recorded by a camera mounted on a vehicle are collected for evaluating the presented algorithms, and the videos are taken on different daytime scenes, including highway, urban common road, urban narrow road, etc. Some roads are covered with japanning, smear, etc. At the first stage, 23,687 samples from the same videos were collected for training and testing, and 17,647 samples were selected randomly for training, including 8,774 vehicle samples (positive samples) and 8,873 non-vehicle samples (negative samples), and the remaining 6,040 samples (denoted as Test data I) for testing which include 4,266 vehicle samples and 1,774 non-vehicle samples. At the second stage, 29,698 samples from different videos with the samples at the first stage were collected for only testing (denoted as Test data II), which include 7, 901 vehicle samples (positive samples) and 4,602 non-vehicle samples (negative samples). The vehicle samples at both the first stage and the second stage include various kinds of vehicles such as cars, trucks and buses as well as different colors such as red, blue, black, gray and white. Furthermore, the vehicle samples include both vehicles near the vehicle mounted with the camera and those that are far. The non-vehicle samples at both stages include roads, buildings, green plants, advertisement boards, bridges, traffic signs, guardrails, and so on. Fig. 6 shows some training examples of vehicle and non-vehicle images, and Fig. 7 shows some test examples of vehicle and non-vehicle images at the second stage.

To evaluate the performance of the approaches, the true positive rate (or vehicle detection rate) t_p and false positive rate f_p were recorded. They are defined in Eq. (10).

$$t_p = \frac{N_{TP}}{N_{TP}+N_{FN}}, \quad f_p = \frac{N_{FP}}{N_{FP}+N_{TN}} \tag{10}$$

where N_{TP} , N_{FP} , N_{TN} and N_{FN} are the numbers of the objects identified as true positives, false positives, true negatives and false negatives respectively. Three experiments are conducted on a PC (CPU: Inter(R) Core2(TM) 2.13GHZ, Memory: 2GB, Operating System: Windows 7, Implementation: Matlab 2012b).

The first experiment aims to validate the performance in classification accuracy of the proposed machine learning method compared to the state-of-the-art ones which perform reasonably well in vehicle classification and for which the code can be obtained or reproduced according to the original papers. The second experiment compares the designed normalization algorithm for the feature vector set to other normalization methods. The third experiment aims to validate the time efficiency of the proposed feature selection algorithm with the AdaBoost compared to the state-of-the-art selection algorithms and the traditional one. All ROIs are normalized to 32×32 grayscale image patches.

In the first experiment, since different datasets will induce different optimal parameters for feature extraction methods and classifiers, we select the optimal parameters in terms of the classification ability. For the feature extraction of PCA [7], we choose the first 79 eigenvectors associated with the first 79 biggest eigenvalues which generate the best classification accuracy. For the feature extraction of Gabor [16], we select 6 angles and 4 orientations. For the feature extraction of wavelet, we select the simplest Haar Wavelet and perform a 6-level decomposition, and then remove the HH part of the first level according to [15]. For the feature extraction of the Gabor combining with wavelet according to [18], the computation of the Gabor features is similar to [16], and the wavelet features is similar to [15]. For the extraction of Haar-like features, 57,519 features were obtained from each 32×32 grayscale image patch [24]. While training the RBF-SVM classifier, we use 5-fold cross-validation to select the best parameters σ and C . While training the cascaded AdaBoost, the vehicle detection accuracy ratio is required to be not smaller than 99.9% and the false alarm (classify non-vehicle to vehicle) ratio is not bigger than 50% at the current stage, and the false alarm (classify non-vehicle to vehicle) ratio of the cascaded classifier is required to be not bigger than 10%, and we select the classifier with the best performance by applying 5-fold cross-validation. While selecting Haar-like features with AdaBoost, we select features by choosing the classifier with the best performance through applying 5-fold cross-validation and select 600 features from 57,519 features. TABLE II shows the evaluation results. Fig. 8 shows the ROCs (Receiver Operating Characteristic Curves) of the seven vehicle detection methods on Test data II.

In addition, two public image data sets are also used to evalute the above machine learning methods. As shown in Table III, the first set is published by Massachusetts Institute of Technology Center for Biological and Computational Learning (MIT CBCL) Group, which consists of the rear- and frontal-viewed vehicle images, and the second set is published by California Institute of Technology (Caltech) Vision Group, which consists of the rear-viewed vehicle images (1999 and 2001 versions). The vehicles in the databases have a wide variety of sizes and in-plane or out-of-plane orientations and are shot against diverse background scenes with different lighting conditions and degrees of occlusion. TABLE IV shows the evaluation results.

In the second experiment, we use three schemes: (1) the statistical normalization [51] [52], (2) min-max normalization [53] and (3) our proposed method to normalize the data. The normalized data as well as the original data are then fed into the RBF-SVM classifier for training and testing. With different attribute normalization schemes, the overall detection results are presented in Fig. 9 and Fig. 10.

TABLE II
EVALUATION RESULTS OF 7 VEHICLE DETECTION METHODS

Methods	Test data I		Test data II	
	t_p	f_p	t_p	f_p
PCA + SVM [7]	96.95%	6.14%	88.74%	5.37%
Gabor+SVM [16]	96.13%	6.54%	90.95%	4.82%
Wavelet + SVM [15]	96.34%	6.43%	87.41%	5.11%
Wavelet + Gabor +SVM [18]	96.81%	5.64%	91.56%	3.98%
Haar-like + Cascaded AdaBoost [46] [47]	97.09%	13.19%	93.66%	11.10%
Haar-like + AdaBoost [24]	97.43%	4.33%	92.28%	3.63%
Proposed method	**97.70%**	**3.44%**	**94.10%**	**3.26%**

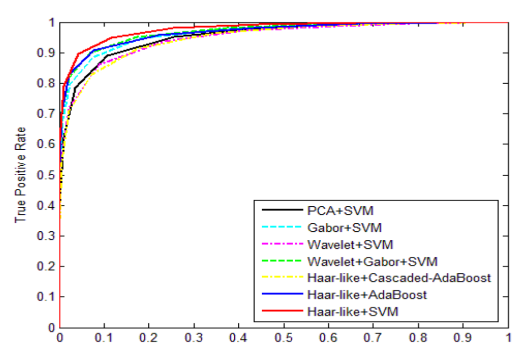


Fig. 8. The ROCs of the seven detection methods on Test data II.

TABLE III
TWO PUBLIC TESTING DATABASES

	MIT CBCL	Caltech rear-viewed vehicle database (1999 and 2001 versions)
Vehicle images	439	652
Image size	128*128 pixels	240*360 pixels

TABLE IV
EVALUATION RESULTS ON THE TWO PUBLIC DATA SETS

Methods	MIT CBCL	Caltech rear-viewed vehicle database (1999 and 2001 versions)
	t_p	t_p
PCA + SVM [7]	88.33%	86.59%
Gabor+SVM [16]	91.46%	90.24%
Wavelet + SVM [15]	91.29%	90.24%
Wavelet + Gabor +SVM [18]	91.81%	91.06%
Haar-like + Cascaded AdaBoost [46] [47]	93.38%	92.38%
Haar-like + AdaBoost [24]	93.55%	92.89%
[54]	---	94.20%
Proposed method	**95.47%**	**94.41%**

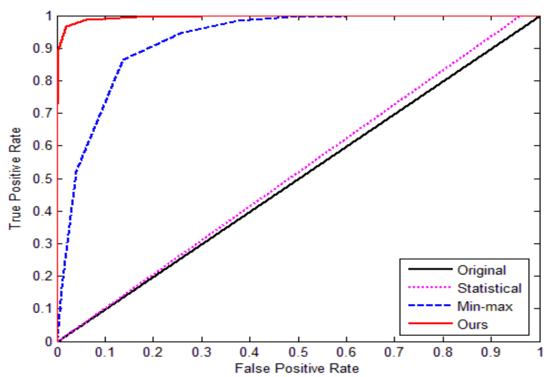


Fig. 9. The ROCs of different normalization methods on Test data I.

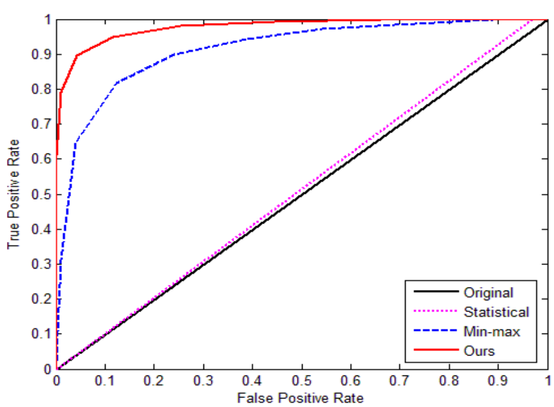


Fig. 10. The ROCs of different normalization methods on Test data II.

TABLE V
EVALUATION RESULTS BEFORE\AFTER THE ADABOOST ALGORITHM BEING IMPROVED

AdaBoost	Training time (hours)
Before improved	117.43$\pm$0.5
Improved [55]	107.27$\pm$0.9
Our improved	101.91$\pm$0.7

In the third experiment, we compare the proposed rapid feature selection algorithm with that in [55] and the traditional one. We conduct the experiment in 5 random trials. In each trial, we randomly divided the training sample set into 5 subsets and perform 5-fold cross-validation. TABLE V shows the mean time as well as the variance of the three methods.

From TABLE II, one can conclude that, compared to the state-of-the-art detection methods, the proposed algorithm produces not only a higher vehicle detection rate (t_p) but also a lower false positive rate (f_p) on both Test data I and Test data II. On Test data II, although the vehicle detection rate of the proposed algorithm is only 0.44% better than that of the method in [46] and [47], but the false positive rate (f_p) of the proposed algorithm is 7.84% lower than that achieved by the method in the literatures. From Fig. 8, one can conclude that the proposed algorithm shows the best performance among all methods.

From TABLE IV, one can conclude that the proposed algorithm shows its superiority on the two public data sets compared to the other methods. In TABLE IV, all methods have better classification results on MIT CBCL than on the Caltech rear-viewed vehicle data set, because most of the vehicle images in MIT CBCL are frontal-viewed vehicles which are more similar to our training samples in distribution.

From Fig. 9 and Fig. 10, one can conclude that, compared to the original data, attribute normalization improves the classification performance significantly, and compared to the other two popular normalization methods in vehicle detection, our improved normalization algorithm is the best choice for RBF-SVM classifier on both Test data I and Test data II. The original data is sensitive to the illumination and easily dominated by the too big attribute values in classification, and the statistical normalization method requires that the attribute data should follow a Normal distribution, which is not always satisfied in real applications. Although the min-max normalization directly used on the original attribute data overcomes the domination of the too big attribute values in classification, it is still sensitive to illumination. The improved normalization method overcomes the above two problems.

From the evaluation in Tables II and IV, it can be observed that the improvement achieved by using the proposed system is only slightly better than that of the state-of-the-art methods. That's because those methods can learn sufficient knowledge from the large scale training dataset effectively and present good performance. The enhanced performance of the proposed system is due to its use of all types of Haar-like features, which improves the tolerance of the vehicle validation process towards geometric variance and partial occlusion, and its application of improved attribute normalization, which reduces the intra-class differences while increasing the inter-class variability and makes the validation process easier.

From TABLE V, one can conclude that the proposed feature selection method saves more than 15 hours in training time than from the traditional one, and saves more than 5 hours compared with the method which relies on the Forward Feature Selection (FFS) algorithm to speed up the feature selection with AdaBoost [55], leading to a more efficient feature selection process.

VII. Conclusion

In this paper, we have proposed a solution based on Haar-like features and RBF-SVM for vehicle detection. Firstly, due to the huge pool of Haar-like features, a fast feature selection algorithm via AdaBoost has been proposed by combining a sample's feature value with its class label. Then, an improved normalization algorithm for feature values has been presented, which can effectively reduce the within-class variation and increase the between-class variability. The experimental results show that the proposed approaches not only speeded up the feature selection process but also showed superiority in vehicle classificaiton ability compared to the state-of-the-art methods.

VIII. Acknowledgements

The authors would like to thank all the anonymous reviewers for their valuable comments.

References

[1] D. Tao, X. Tang, X. Li, X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval", IEEE Trans. Pattern Anal. Mach. Intell, vol. 28, no. 7, pp. 1088-1099, 2006.

[2] W. Liu, D. Tao, "Multiview hessian regularization for image annotation", IEEE Transactions on Image Processing, vol. 22, no. 7, pp. 2676-2687, 2013.

[3] F. Zhu and L. Shao,"Weakly-supervised cross-domain dictionary learning for visual recognition", International Journal of Computer Vision (IJCV), vol. 109, no. 1-2, pp. 42-59, Aug. 2014.

[4] S. Sivaraman, M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis", IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pp.1773-1795, 2013.

[5] J. Li, D. Tao, "Simple exponential family PCA", IEEE Trans. Neural Netw. Learning Syst., vol. 24, no. 3, pp. 485-497, 2013.

[6] D. Tao, X. Li, X. Wu, S. J. Maybank, "Geometric mean for subspace selection", IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 2, pp. 260-274 , 2009.

[7] J. Wu and X. Zhang, "A PCA classifier and its application in vehicle detection", Proc. IEEE Int'l Joint Conf. Neural Networks, vol.1, pp. 600 – 604, 2001.

[8] T. Kato, Y. Ninomiya, and I. Masaki, "Preceding vehicle recognition based on learning from sample images", IEEE Trans. Intell. Transp. Syst., vol. 3, no. 4, pp. 252 – 260, Dec. 2002.

[9] N. Matthews, P. An, D. Charnley, and C. Harris, "Vehicle detection and recognition in greyscale imagery", Control Eng. Pract., vol. 4, no. 4, pp. 473 – 479, Apr. 1996.

[10] S. L. Phung, D. Chai, and A. Bouzerdoum, "A distribution-based face/non-face classification technique", Aust. J. Intell. Inf. Process. Syst., vol. 7, no. 3/4, pp. 132 – 138, 2001.

[11] A. N. Rajagopalan, P. Burlina, and R. Chellapa, "Higher order statistical learning for vehicle detection in images", in Proc. IEEE Int. Conf. Comput. Vis., 1999, vol. 2, pp. 1204 – 1209.

[12] Z. Sun, G. Bebis, and R. Miller, "Object detection using feature subset selection", Pattern Recognit., vol. 37, no. 11, pp. 2165 – 2176, Nov. 2004.

[13] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection", IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 1, pp. 39 – 51, Jan. 1998.

[14] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection", Int'l J. Computer Vision, vol. 38, no 1, pp. 15-33, 2000.

[15] Z. Sun, G. Bebis, and R. Miller, "Quantized wavelet features and support vector machines for on-road vehicle detection", 7th International Conference on Control, Automation, Robotics and Vision, 2002, pp. 1641-1646.

[16] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using Gabor filters and support vector machines", International Conference on Digital Signal Processing, 2002, pp. 1019-1022.

[17] D. Tao, X. Li, X. Wu, S.J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition", IEEE Trans. Pattern Anal. Mach. Intell, vol. 29, no. 10, pp. 1700-1715, 2007.

[18] Z. Sun, G. Bebis, and R. Miller, "Improving the performance of on-road vehicle detection by combining Gabor and wavelet features", The IEEE 5th International Conference on Intelligent Transportation Systems, 2002, pp. 130-135.

[19] D. Lowe, "Object recognition from local scale-invariant features", in Proc. Int. Conf. Comput. Vis., 1999, pp. 1150 – 1157.

[20] X. Zhang, N. Zheng, Y. He, and F. Wang, "Vehicle detection using an extended hidden random field model", in Proc. 14th Int. IEEE Conf. ITSC, Oct. 2011, pp. 1555 – 1559.

[21] M. Cheon, W. Lee, C. Yoon, and M. Park, "Vision-based vehicle detection system with consideration of the detecting location", IEEE Trans. Intell. Transp. Syst., vol. 13, no. 3, pp. 1243 – 1252, 2012.

[22] B.F. Lin, Y.M. Chan, L.C. Fu, P.Y. Hsiao, L.A. Chuang, S.S. Huang, and M.-F. Lo, "Integrating appearance and edge features for sedan vehicle detection in the blind-spot area", IEEE Trans. Intell. Transp. Syst., vol. 13, no. 2, pp. 737 – 747, Jun. 2012.

[23] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features", Comput. Vis. Image Understand., vol. 110, no. 3, pp. 346 – 359, 2008.

[24] X. Wen, Y. Zheng, "An improved algorithm based on AdaBoost for vehicle recognition", The 2nd International Conference on Information Science and Engineering (ICISE2010), Hangzhou, China, 2010, pp. 4-7.

[25] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features", In IEEE Conference on Computer Vision and Pattern Recognition, January 2001, pp. 511 – 518.

[26] P. Viola, M. Jones, "Robust real-time face detection", International Journal of Computer Vision", vol. 57, no. 2, pp. 137 - 154, 2004.

[27] P. Viola and M. Jones, "Robust real-time object detection", In IEEE ICCV Workshop on Statistical and Computational Theories of Vision, Vancouver, Canada, July 2001, pp. 1-30.

[28] R. Miller, Z. Sun and G. Bebis, "Monocular precrash vehicle detection: Features and classifiers", IEEE Trans. Image Process., vol. 15, no. 7, pp. 2019 – 2034, Jul. 2006.

[29] O. Ludwig and U. Nunes, "Improving the generalization properties of neural networks: An application to vehicle detection", in Proc. 11th Int. IEEE Conf. ITSC, Oct. 2008, pp. 310 – 315.

[30] X. Wen , H. Zhao, et al, "A rear-vehicle detection system for static images based on monocular vision", 9th International Conference on Control, Automation, Robotics and Vision, Singapore, March 2006, pp. 2421 – 2424.

[31] W. Liu, X. Wen, B. Duan, et al, "Rear vehicle detection and tracking for lane change assist", Proceedings of the IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 2007, pp. 252 - 257.

[32] S. Teoh and T. Brunl (2012, Sep.). "Symmetry-based monocular vehicle detection system", Mach. Vis. Appl. [Online]. 23(5), pp. 831 – 842. Available: http://dx.doi.org/10.1007/s00138-011-0355-7.

[33] S. Sivaraman and M. M. Trivedi, "Active learning for on-road vehicle detection: A comparative study", Mach. Vis. Appl.—Special Issue Car Navigation and Vehicle Systems, pp. 1 – 13, Dec. 2011.

[34] Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Learning a family of detectors via multiplicative kernels", IEEE Trans. Pattern Anal.Mach. Intell., vol. 33, no. 3, pp. 514 – 530, Mar. 2011.

[35] N. Blanc, B. Steux, and T. Hinz, "LaRASideCam: A fast and robust vision-based blindspot detection system", in Proc. IEEE Intell. Veh. Symp., Jun. 2007, pp. 480 – 485.

[36] Z. Kim, "Realtime obstacle detection and tracking based on constrained Delaunay triangulation", in Proc. IEEE ITSC, Sep. 2006, pp. 548 – 553.

[37] Y. Zhang, S. Kiselewich, andW. Bauson, "Legendre and Gabor moments for vehicle recognition in forward collision warning", in Proc. IEEE ITSC, Sep. 2006, pp. 1185 – 1190.

[38] T. Liu, N. Zheng, L. Zhao, and H. Cheng, "Learning based symmetric features selection for vehicle detection", in Proc. IEEE Intell. Veh. Symp., Jun. 2005, pp. 124 – 129.

[39] A. Khammari, F. Nashashibi, Y. Abramson, and C. Laurgeau, "Vehicle detection combining gradient analysis and AdaBoost classification", in Proc. IEEE Intell. Transp. Syst., Sep. 2005, pp. 66 – 71.

[40] J. Cui, F. Liu, Z. Li, and Z. Jia, "Vehicle localization using a single camera", in Proc. IEEE IV, Jun. 2010, pp. 871 – 876.

[41] D. Withopf and B. Jahne, "Learning algorithm for real-time vehicle tracking", in Proc. IEEE ITSC, Sep. 2006, pp. 516 – 521.

[42] I. Kallenbach, R. Schweiger, G. Palm, and O. Lohlein, "Multi-class object detection in vision systems using a hierarchy of cascaded classifiers", in Proc. IEEE Intell. Veh. Symp., 2006, pp. 383 – 387.

[43] T. Son and S. Mita, "Car detection using multi-feature selection for varying poses", in Proc. IEEE Intell. Veh. Symp., Jun. 2009, pp. 507 – 512.

[44] D. Acunzo, Y. Zhu, B. Xie, and G. Baratoff, "Context-adaptive approach for vehicle detection under varying lighting conditions", in Proc. IEEE ITSC, 2007, pp. 654 – 660.

[45] C. Szegedy, A. Toshev and E. Erhan, "Deep neural network for object detection", In Advances in Neural Information Processing Systems (NIPS) , 2013, pp. 2553-2561.

[46] R. Lienhart, J. Maydt, "An extended set of Haar - like features for rapid object detection", The IEEE International Conference on Image Processing, January 2002, pp. 900 - 903.

[47] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection", Proceedings of the 25th German Pattern Recognition Symposium, 2003, pp. 297-304.

[48] Y. Freund, R. E. Schapire, "Experiments with a new boosting algorithm", In Proceedings of the 13th Conference on Machine Learning, 1996, pp. 148-156.

[49] V. Vapnik, "The Nature of Statistical Learning Theory", Springer Verlag, 1995.

[50] C. Burges, "Tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 955-974, 1998.

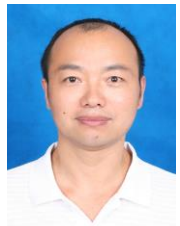
[51] Y. Li, B. Fang, L. Guo, Y. Chen, "Network anomaly detection based on tcm-knn algorithm", In: ASIACCS, 2007, pp. 13 – 19.

[52] W. Ma, D. Tran, D. Sharma, "A study on the feature selection of network traffic for intrusion detection purpose", In: ISI, 2008, pp.245 – 247.

[53] Y. Liao, V.R.Vemuri, A. Pasos, "Adaptive anomaly detection with evolving connectionist systems", Network and Computer Applications, Vol. 30, No. 1, pp. 60 – 80, 2007.

[54] C Wang and, and J Lien, "Automatic Vehicle Detection Using Local Features—A Statistical Approach", IEEE Transactions on Intelligent Transportation System, vol. 9, no. 1, pp. 83–96, Mar. 2008.

[55] J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg, "Fast asymmetric learning for cascade face detection", IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 30, no. 3, pp. 369–382, Mar. 2008.



Xuezhi Wen received the Ph. D. degree in computer application technique from Northeastern University, Shenyang, China in 2008.

He is currently an Associate Professor with Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, China, and also an Associate Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, China. He is a member of ACM. His research interests include pattern recognition, image processing and intelligent transportation.

Ling Shao (M'09–SM'10) received the B.Eng. degree in Electronic and Information Engineering from the University of Science and Technology of China (USTC), the M.Sc. degree in Medical Image Analysis and the Ph.D. (D.Phil.) degree in Computer Vision at the Robotics Research Group from the University of Oxford.

He is currently a Full Professor with the Department of Computer Science and Digital Technologies, Northumbria University, Newcastle, UK. Previously, he was a Senior Lecturer (2009-2014) with the Department of Electronic and Electrical Engineering at the University of Sheffield and a Senior Scientist (2005-2009) with Philips Research, The Netherlands. His research interests include Computer Vision, Image/Video Processing, Pattern Recognition and Machine Learning. He has authored/co-authored over 130 academic papers in refereed journals and conference proceedings and over 10 EU/US patents. Ling Shao has been an associate (or guest) editor of IEEE Transactions on Cybernetics, Information Sciences, Pattern Recognition, IEEE Transactions on Neural Networks and Learning Systems and several other journals. He has organized several workshops with top conferences, such as ICCV, ACM Multimedia and ECCV. He has been serving as a Program Committee member for many international conferences, including ICCV, CVPR, ECCV, ACM MM, and so on. He is a Fellow of the British Computer Society, a Fellow of the IET and a Senior Member of the IEEE.

Wei Fang received the Ph.D. degree in computer science from the SooChow University, China, in 2009.

He is currently an Associate Professor in the Jiangsu Engineering Center of Network Monitoring at the Nanjing University of Information Science & Technology in China. He is a senior member of CCF, and a member of ACM. His current research interests are in the areas of Data Mining, Big Data Analytics and Cloud Computing.

Yu Xue received the Ph.D. degree in college of computer science and technology, Nanjing University of Aeronautics and Astronautics, China, in 2013.

He is currently a Lecturer with Jiangsu Engineering Center of Network Monitoring at Nanjing University of Information Science and Technology in China, and also a Lecturer with School of Computer and Software, Nanjing University of Information Science and Technology, China. He is a member of IEEE and CCF. His research interests include Computational Intelligence, Electronic Countermeasure and Internet of Things.